

AMENDMENTS TO THE CLAIMS

- 1-24. (Canceled)
25. (New) A method of constructing a hierarchical database from an initial plurality of rules, the method comprising:
- adding a first rule of the initial plurality of rules to a first sub-database if a first bit of the rule is a logic '0' value;
- adding the first rule to a second sub-database if the first bit is a logic '1' value; and
- adding the first rule to a third sub-database if the first bit is in a masked state, 'X', indicating that the first bit may be either a logic '1' or a logic '0' value.
26. (New) The method of claim 25 further comprising adding each other rule of the initial plurality of rules to either the first sub-database, the second sub-database or the third sub-database, according to a first bit of the other rule.
27. (New) The method of claim 26 wherein the first sub-database, second sub-database and third sub-database form a first hierarchical level in the hierarchical database and wherein the method further comprises, after adding the first rule and each other rule of the initial plurality of rules to either the first sub-database, the second sub-database or the third sub-database, splitting the first sub-database into a second hierarchical level of sub-databases if, before splitting the first sub-database, the first sub-database contains more than a predetermined number of rules.
28. (New) The method of claim 27 wherein splitting the first sub-database into a second

hierarchical level of sub-databases comprises adding a first rule of the plurality of rules within the first sub-database to:

a fourth sub-database if a second bit of the rule is a logic '0' value;

a fifth sub-database if the second bit of the rule is a logic '1' value; and

a sixth sub-database if the second bit of the rule is in the masked state, 'X'.

29. (New) The method of claim 28 further comprising splitting the second sub-database into a corresponding second hierarchical level of sub-databases if the second sub-database comprises more than the predetermined number of rules, and splitting the third sub-database into a corresponding second hierarchical level of sub-databases if the third sub-database comprises more than the predetermined number of rules.
30. (New) The method of claim 29 further comprising repeatedly splitting each sub-database at the second or lower hierarchical level into respective sub-databases at an even lower hierarchical level until all the sub-databases have no more than the predetermined number of rules.
31. (New) The method of claim 29 wherein splitting sub-databases at each hierarchical level comprises splitting the sub-databases based on a logical value of a bit at a bit position different than bit positions evaluated at higher hierarchical levels.
32. (New) The method of claim 30 wherein each rule appears in no more than one of the sub-databases.
33. (New) The method of claim 30 further comprising merging a sub-database at the second or

lower hierarchical level into a respective sub-database at a higher hierarchical level if the sub-database has fewer than a minimum number of rules.

34. (New) The method of claim 33 wherein the minimum number of rules is one-third the predetermined number of rules.
35. (New) A method of searching a hierarchical database, the method comprising:
selectively searching a first set of sub-databases if a first bit of a received packet header is a logic '0' value;
selectively searching a second set of sub-databases if the first bit is a logic '1' value; and
selectively searching a third set of sub-databases if the first bit is in a masked state, 'X',
indicating that the first bit may be either a logic '1' or a logic '0' value.
36. (New) The method of claim 35 wherein selectively searching one of the first, second and third sets of sub-databases comprises selectively searching a subset of the respective set of sub-databases based on one or more other bits of the packet header.
37. (New) The method of claim 36 wherein selectively searching a subset of the respective set of sub-databases comprises:
selectively searching a first subset of the set of sub-databases if a second bit of the packet header is a logic '0' value;
selectively searching a second subset of the set of sub-databases if the second bit is a logic '1' value; and
selectively searching a third subset of the set of sub-databases if the second bit is in a masked state, 'X'.

38. (New) The method of claim 37 wherein selectively searching a subset of the respective set of sub-databases comprises searching a first sub-database of the respective set of sub-databases based on a ternary match of the bits of the packet header.
39. (New) An apparatus comprising:
an interface to receive a plurality of rules; and
circuitry to organize the plurality of rules into a hierarchical database of rules, including
circuitry to store a first rule of the plurality of rules in either a first sub-database, a second sub-database, or a third sub-database according to whether a first bit of the first rule is a logic '0' value, a logic '1' value or in a masked state, 'X', respectively, the masked state indicating that the first bit may be either a logic '1' or a logic '0' value.
40. (New) The apparatus of claim 39 wherein the first, second and third sub-databases form a first hierarchical level of the hierarchical database.
41. (New) The apparatus of claim 39 wherein each of the first, second and third sub-databases can store at most, a predetermined number of rules.
42. (New) The apparatus of claim 39 further comprising circuitry to store the first rule in either a fourth sub-database, a fifth sub-database, or a sixth sub-database according to whether a second bit of the first rule is a logic '0' value, a logic '1' value or in a masked state, 'X', respectively.
43. (New) The apparatus of claim 41 wherein the fourth, fifth and sixth sub-databases form a

second hierarchical level of the hierarchical database.

44. (New) An apparatus comprising:

means for adding a first rule to a first sub-database if a first bit of the rule is a logic '0' value;

means for adding the first rule to a second sub-database if the first bit is a logic '1' value;
and

means for adding the first rule to a third sub-database if the first bit is in a masked state, 'X', indicating that the first bit may be either a logic '1' or a logic '0' value.